IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nicole Laforest
15/11/2022

Outline

Executive Summary

Summary of methodologies	Summary of all results
• Data Collection via API, Web Scraping • Exploratory Data Analysis (EDA) with Data Visualization • EDA with SQL • Interactive Map with Folium • Dashboards with Plotly Dash • Predictive Analysis	• Exploratory Data Analysis Results • Interactive maps and dashboard • Predictive Results

Introduction

Project background and context

- With these project we want to predict if the Falcon 9 first stage will successfully land.

Problems you want to find answers

- The main characteristics of a successful or failed landing
- The effects of each relationship of the rocket variables on the success or failure of a landing
- The conditions which will allow SpaceX to achieve the best landing success rate

Section 1

Methodology

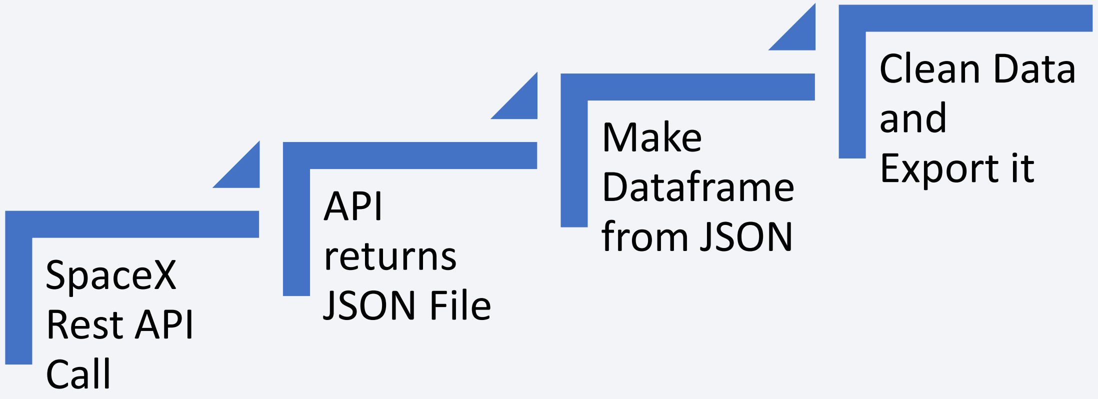
Methodology

Executive Summary

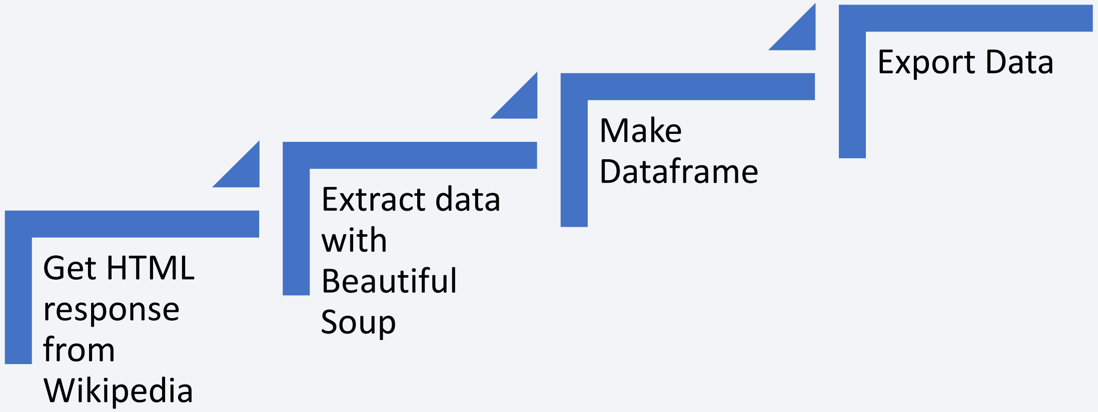
- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Dropping unnecessary columns
  - One Hot Encoding for classification models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Datasets are collected from Rest SpaceX API and webscrapping Wikipedia
  - The information obtained by the API are rocket, launches, payload information.
    - The Space X REST API URL is api.spacexdata.com/v4/



- The information obtained by the webscrapping of Wikipedia are launches, landing, payload information.
  - URL is https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922



Data Collection – SpaceX API

Getting Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

Convert Response to JSON File

```
data = response.json()
data = pd.json_normalize(data)
```

Transform Data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
getBoosterVersion(data)
```

Create Dictionary with Data

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

Create dataframe

```
data = pd.DataFrame.from_dict(launch_dict)
```

Filter Dataframe

```
data_falcon9 = data[data['BoosterVersion']!='Falcon 1']
```

Export File

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

Getting Response from HTML

```
response = requests.get(static_url)
```

Create BeautifulSoup Object

```
soup = BeautifulSoup(response.text, "html5lib")
```

Find All Tables

```
html_tables = soup.findAll('table')
```

Get Column Names

```
for th in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0 :
        column_names.append(name)
```

Create dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Add data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all
    # get table row
     for rows in table.find_all("tr"):
        #check to see if first table heading is a
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.stri
                flag=flight_number.isdigit()
```

Create dataframe from dictionary

```
df=pd.DataFrame(launch_dict)
```

Export to Fila

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

Calculate launches number for each site

```
df['LaunchSite'].value_counts()

CCAFS SLC 40     55
KSC LC 39A       22
VAFB SLC 4E      13
Name: LaunchSite, dtype: int64
```

Calculate the number and occurence of each orbit

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes

True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
None ASDS       2
False Ocean     2
False RTLS      1
Name: Outcome, dtype: int64
```

Calculate number and occurence of misión outcome per orbit type

```
df['Orbit'].value_counts()

GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
SO        1
ES-L1     1
HEO       1
GEO       1
Name: Orbit, dtype: int64
```

Create landing outcome lamebl from Outcom column

```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

Export to File

```
df.to_csv("dataset_part_2.csv", index=False)
```

EDA with Data Visualization

Scatter Graphs

- Flight Number vs. Payload Mass
- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Orbit vs. Flight Number
- Payload vs. Orbit Type
- Orbit vs. Payload Mass
- **show relationship between variables. This relationship is called the correlation**

Bar Graph

- Success rate vs orbit
- **Show the relationship between numeric and categoric variables.**

Line Graph

- Success Rate vs Year
- **Show data variables and their trends. Line graphs can help to show global behavior and make prediction for unseen data**

EDA with SQL

We performed SQL queries to gather and understand data from dataset for display

- All Launch Site Names
- Launch Site Names Begin with 'CCA'
- Total Payload Mass
- Average Payload Mass by F9 v1.1
- First Successful Ground Landing Date
- Successful Drone Ship Landing with Payload between 4000 and 6000
- Total Number of Successful and Failure Mission Outcomes
- Boosters Carried Maximum Payload
- 2015 Launch Records
- Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

Build an Interactive Map with Folium

Folium map object is a map centered on NASA Johnson Space Center at Houson, Texas

Red circle at NASA Johnson Space Center's coordinate with label showing its name	• folium.Circle, folium.map.Marker
Red circles at each launch site coordinates with label showing launch site name	• folium.Circle, folium.map.Marker, folium.features.DivIcon
The grouping of points in a cluster to display multiple and different information for the same coordinates	• folium.plugins.MarkerCluster
Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing.	• folium.map.Marker, folium.Icon
Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them.	• folium.map.Marker, folium.PolyLine, folium.features.DivIcon

These objects were created to understand the problem and the data. We can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings.

Build a Dashboard with Plotly Dash

We add to the dashboard:

Dropdown:

- Allows a user to choose the launch site or all launch sites
- dash_core_components.Dropdown

Pie chart

- Shows the total success and the total failure for the launch site chosen with the dropdown component
- plotly.express.pie

Rangeslider

- Allows a user to select a payload mass in a fixed range
- dash_core_components.RangeSlider

Scatter chart

- Shows the relationship between two variables, in particular Success vs Payload Mass
- plotly.express.scatter

Predictive Analysis (Classification)

Data preparation

- Load dataset
- Normalize data
- Split data into training and test sets.

Model preparation

- Selection of machine learning algorithms
- Set parameters for each algorithm to GridSearchCV
- Training GridSearchModel models with training dataset

Model evaluation

- Get best hyperparameters for each type of model
- Compute accuracy for each model with test dataset
- Plot Confusion Matrix

Model comparison

- Comparison of models according to their accuracy
- The model with the best accuracy will be chosen

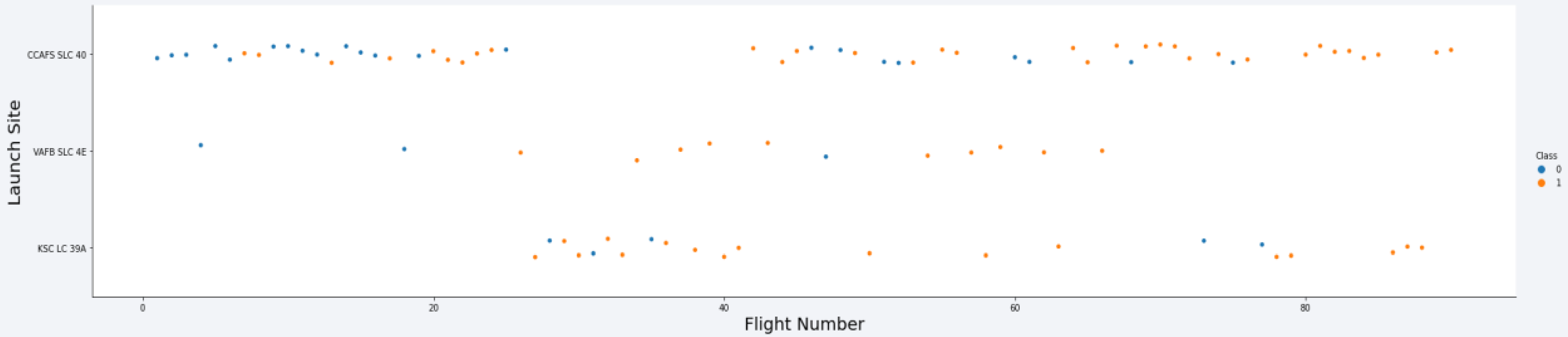
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

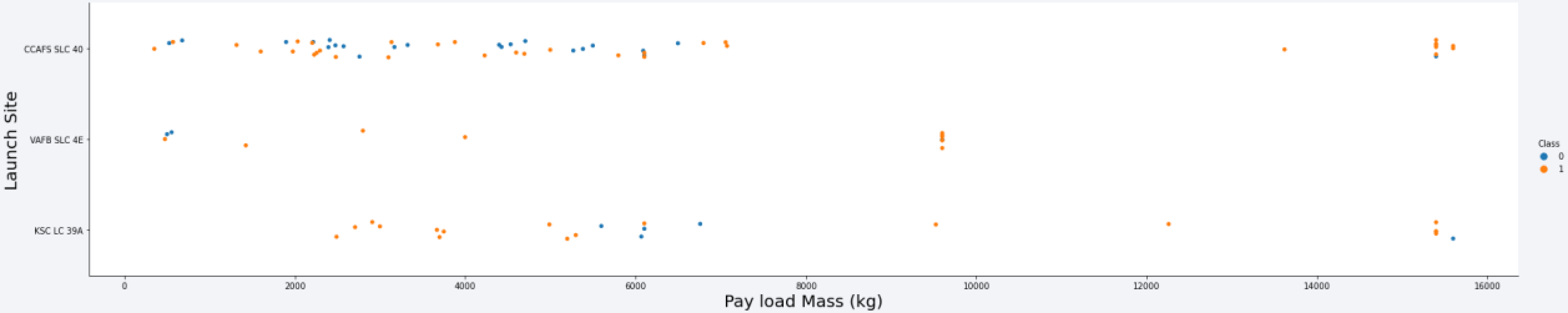
Insights drawn from EDA

Flight Number vs. Launch Site



The success rate is increasing for each site

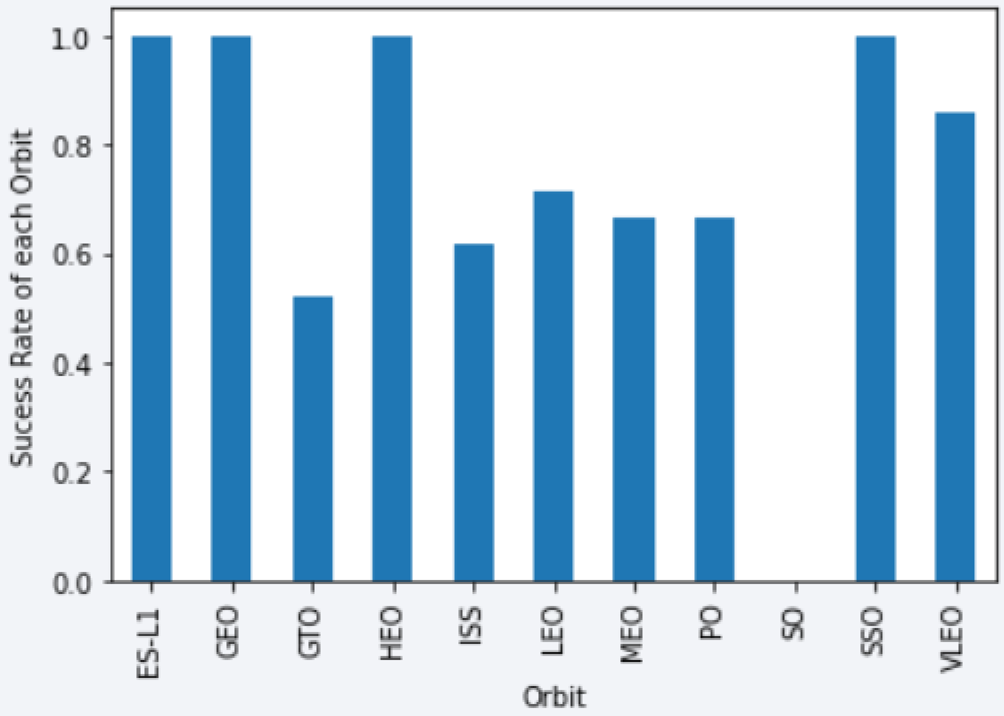
Payload vs. Launch Site



Each launch site, accept different payload. A heavier, may be a consideration for a successful landing.

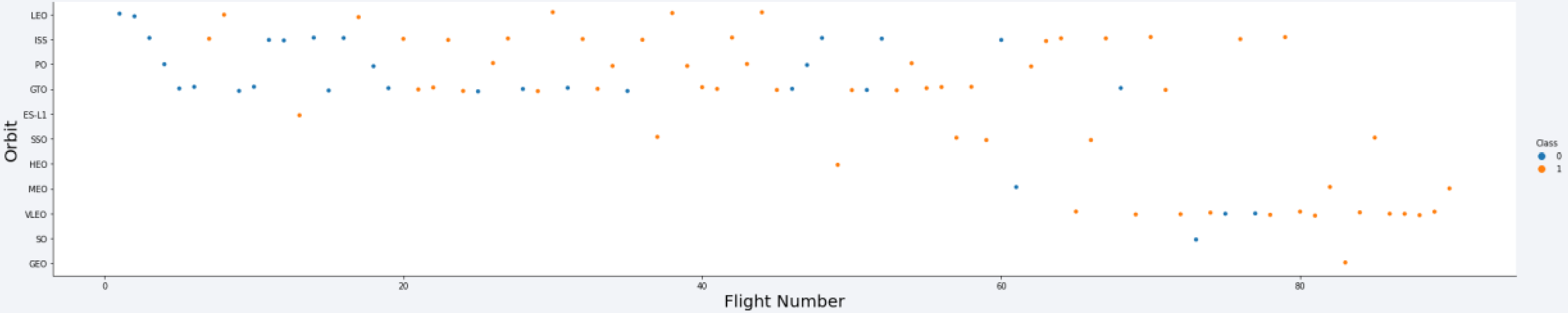
But, a too heavy can make a landing fail

Success Rate vs. Orbit Type



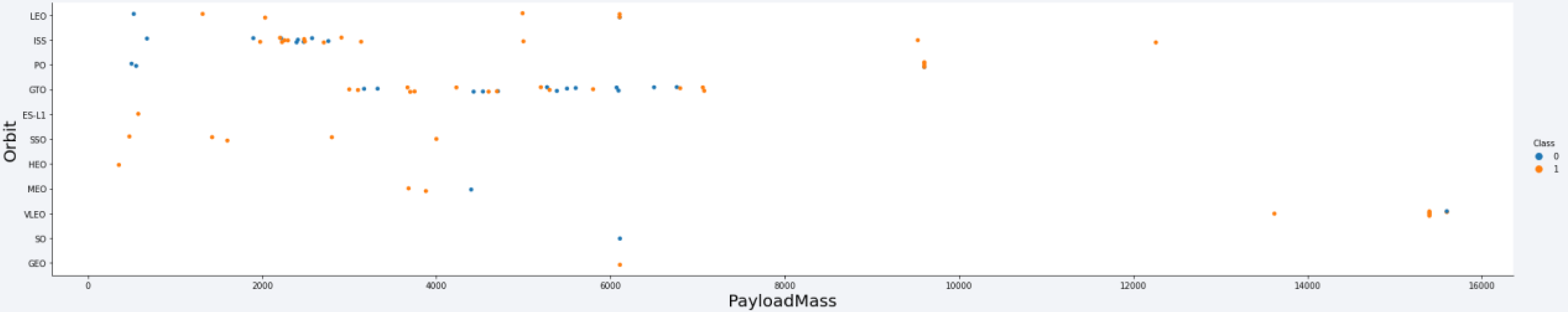
ES-L1, GEO, HEO, SSO have the best success rate

Flight Number vs. Orbit Type



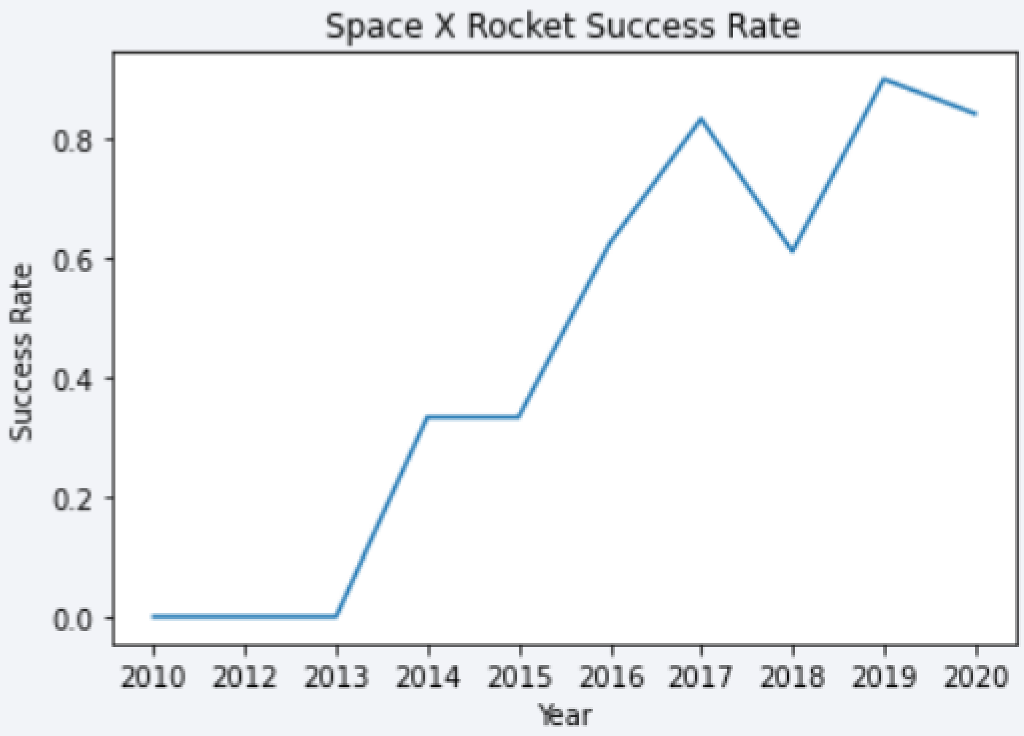
- **Success rate increases with the number of flights for the LEO orbit.**
- **There is no relation between the success rate and the number of flights for GTO**
- **SSO or HEO have high success rate**

Payload vs. Orbit Type



- **The weight of the payloads have big influence on the success rate of the launches in certain orbits**
- **Heavier payloads improve the success rate for the LEO orbit**
- **Decreasing the payload weight for a GTO orbit improves the success of a launch**

Launch Success Yearly Trend



- **There is an increase in the Space X Rocket success rate since 2013**

All Launch Site Names

SQL

Results

Explanation

```
SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

DISTINCT remove duplicate LAUNCH_SITE.

Launch Site Names Begin with 'CCA'

SQL

Results

Explanation

```
SELECT * FROM SPACEXTBL WHERE "LAUNCH_SITE" LIKE '%CCA%' LIMIT 5
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

WHERE - LIKE filters launch sites that contain the substring CCA.
LIMIT 5 shows 5 records from filtering

Total Payload Mass

SQL

```
SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
```

Results

SUM("PAYLOAD_MASS__KG_")
45596

Explanation

SUM shows the total payload carried
FROM: allows to make the filter

Average Payload Mass by F9 v1.1

SQL

```
SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'
```

Results

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

Explanation

AVG: average payload mass carried
FROM: allows filter where the booster version contains the substring F9 v1.1.

First Successful Ground Landing Date

SQL

```
SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'
```

Results

MIN("DATE")
01-05-2017

Explanation

WHERE filters dataset in order to keep only records where landing was successful. MIN function, we select the record with the oldest date.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL

```
%sql SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Success (drone ship)' \
AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

Results

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Explanation

This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg.
The WHERE and AND clauses filter the dataset.

Total Number of Successful and Failure Mission Outcomes

SQL

```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, \
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

Results

SUCCESS	FAILURE
100	1

Explanation

SELECT show the subqueries that return results. The first subquery counts the successful mission. The second subquery counts the unsuccessful mission. The WHERE clause followed by LIKE clause filters mission outcome.
COUNT: counts records filtered

Boosters Carried Maximum Payload

SQL

Results

Explanation

```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

MAX: return only the heaviest payload mass
SELECT DISTINCT: returns unique booster version with the heaviest payload mass

2015 Launch Records

SQL

```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

Results

MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

Explanation

Substr function process date in order to take month or year.
Substr(DATE, 4, 2) shows month.
Substr(DATE,7, 4) shows year.

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

SQL

```
%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING _OUTCOME" \
ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```

Results

Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

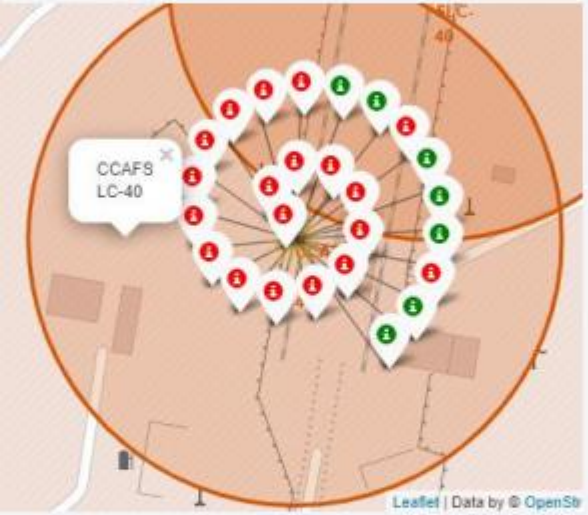
Explanation

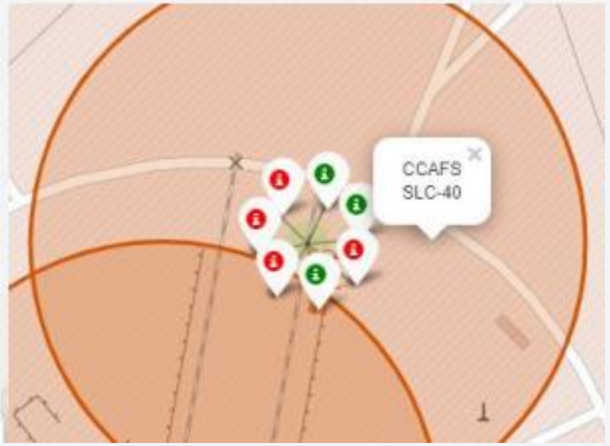
GROUP BY clause groups results by landing outcome
ORDER BY COUNT DESC shows results in decreasing order

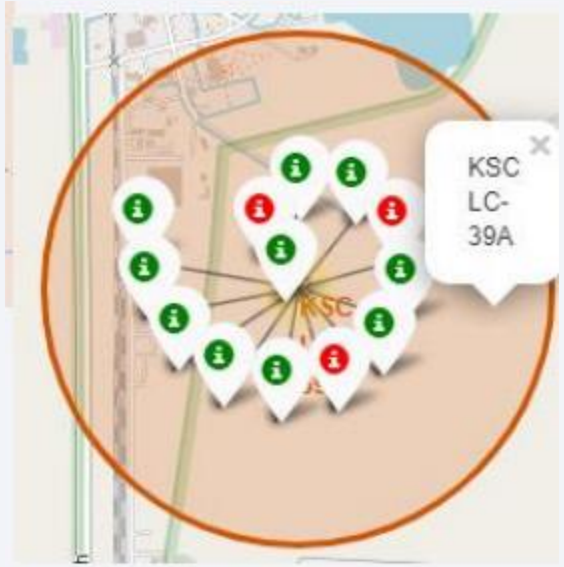
Section 3

Launch Sites Proximities Analysis

Folium map – Color Labeled Markers for launches

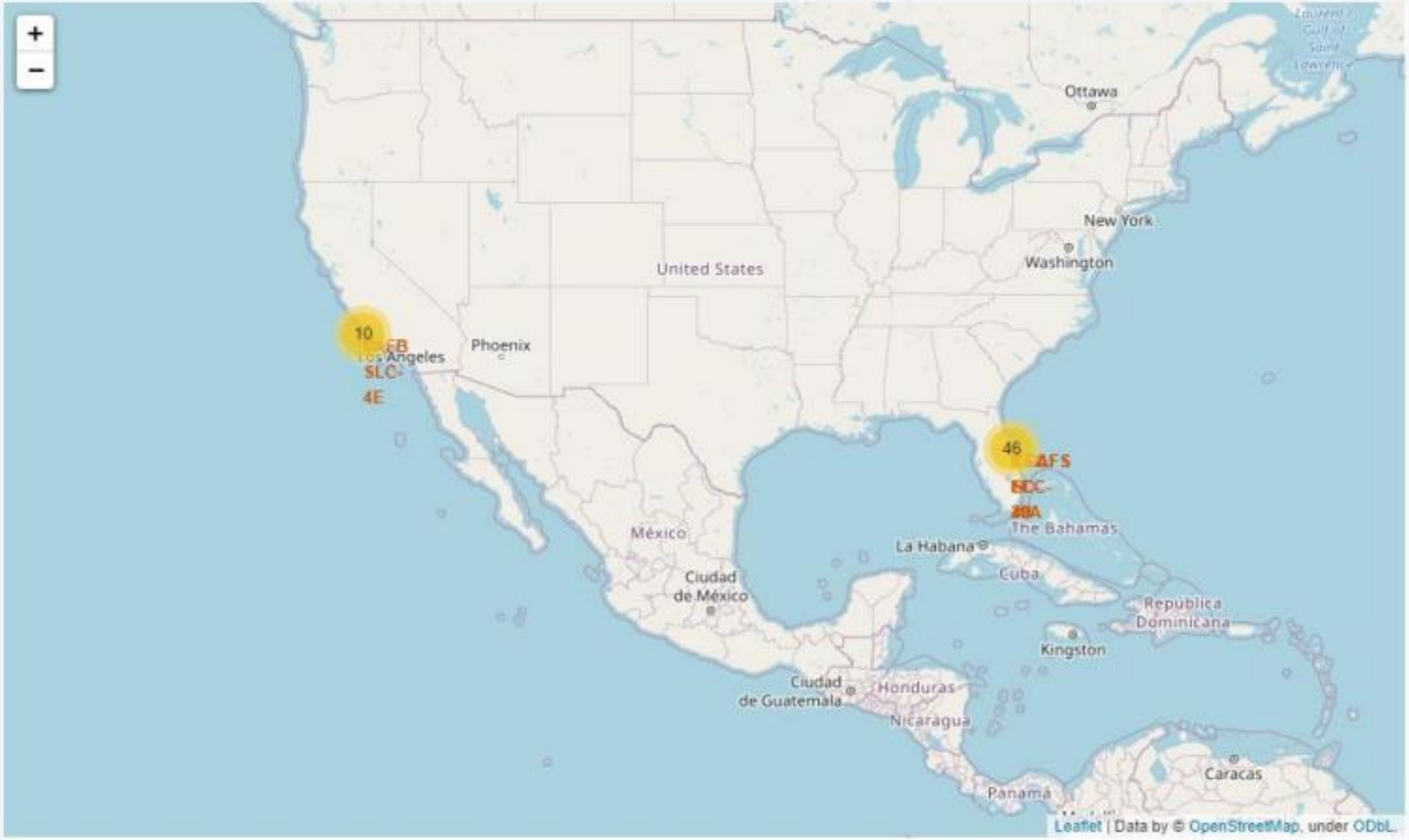






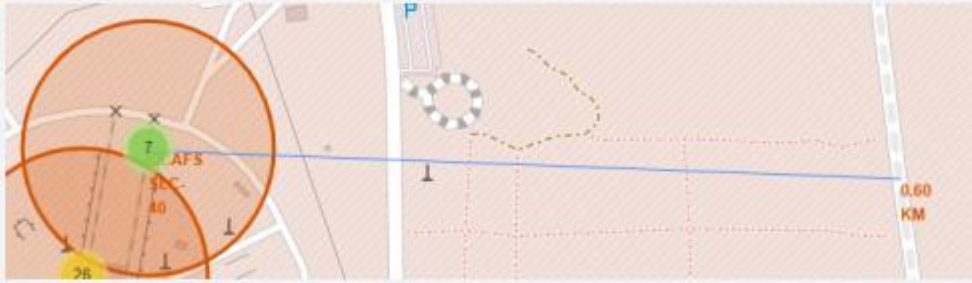
Green marker represents successful launches.
Red marker represents unsuccessful launches.
KSC LC-39A has a higher launch success rate.

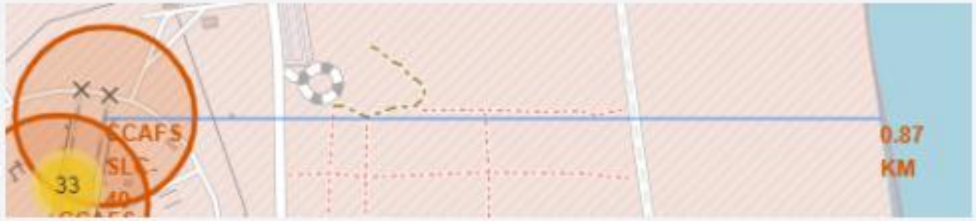
Folium map – Location of the Ground stations



All the Space X launch sites are located on the coast of the United States

Folium Map – Distances with CCAFS SLC-40



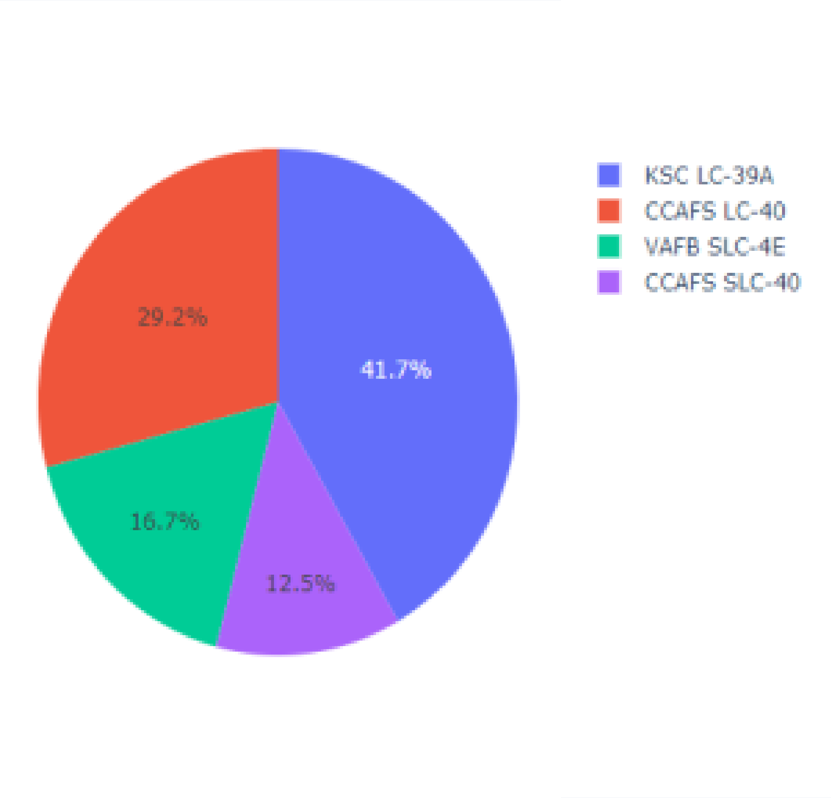


CCAFS SLC-40 in in close proximity to railways
CCAFS SLC-40 is in close proximity to highways
CCAFS SLC-40 is in close proximity to coastline
CCAFS SLC-40 don’t keeps certain distance away from cities

Section 4

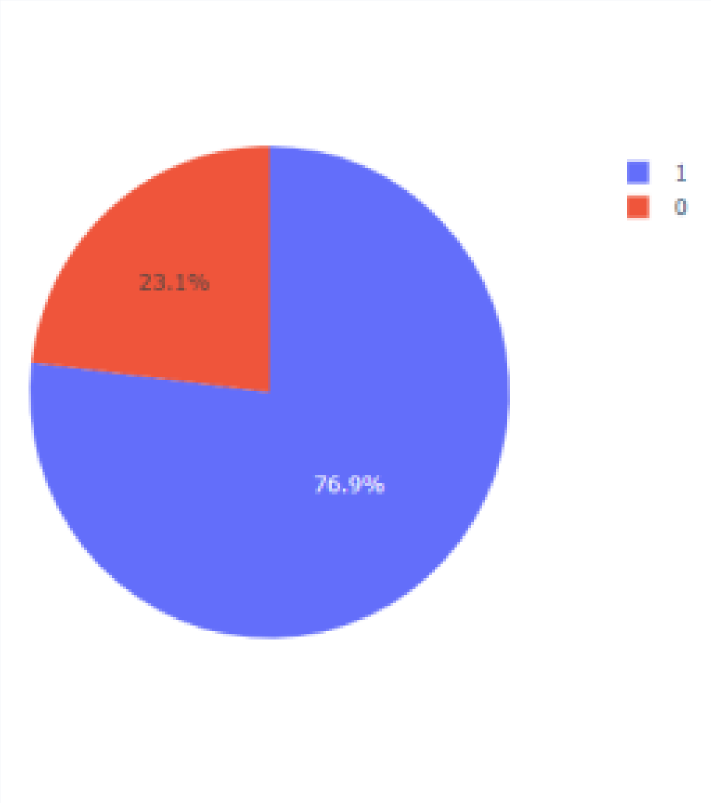
Build a Dashboard with Plotly Dash

Dashboard – Success by Site



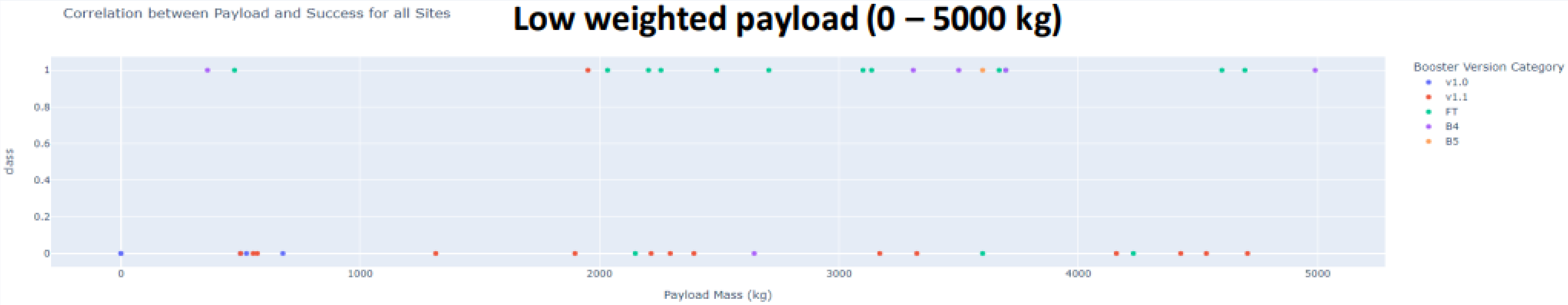
KSC LC-39A has the best success rate of launches

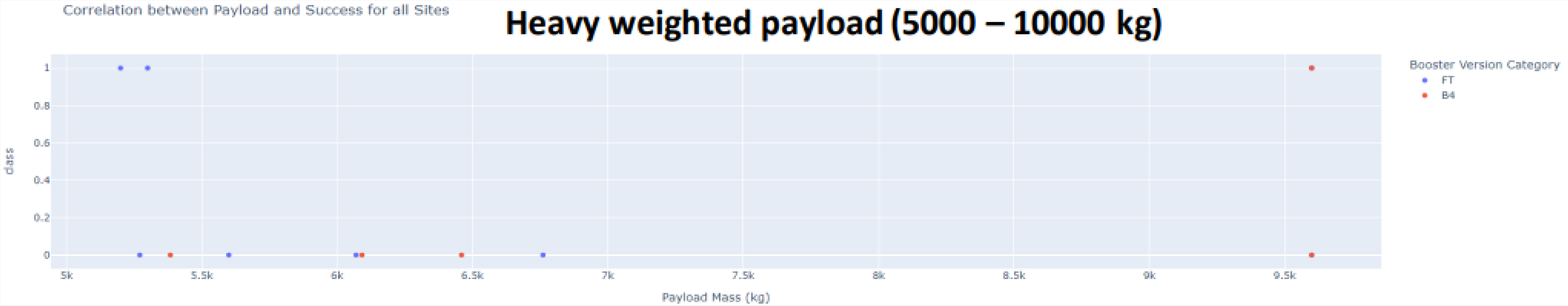
Dashboard – Success launches for Site KSC LC-39A



KSC LC-39A has achieved a 76.9% and a 23.1% failure rate.

Dashboard – Payload mass vs Outcome for all sites with different payload mass selected





Low weighted payloads have a better success rate than the heavy weighted payloads.

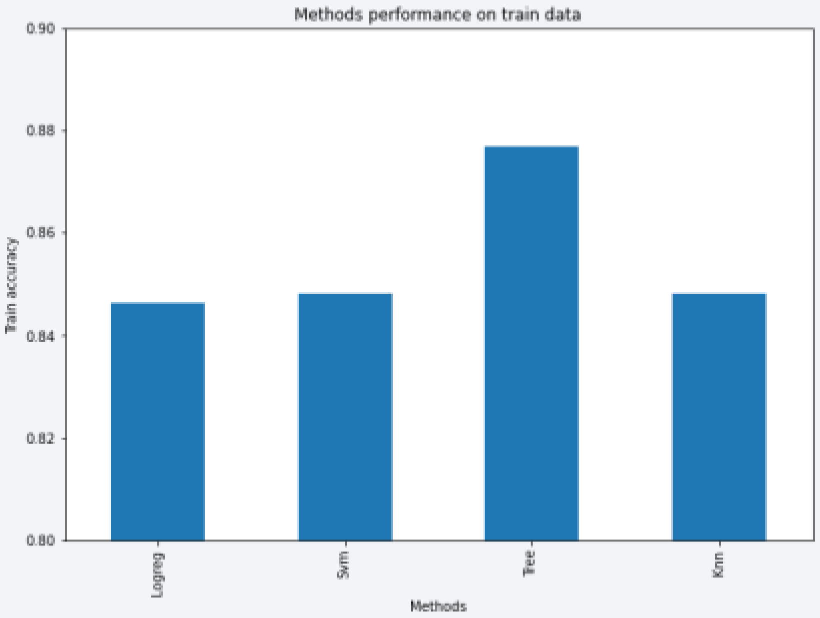
Section 5

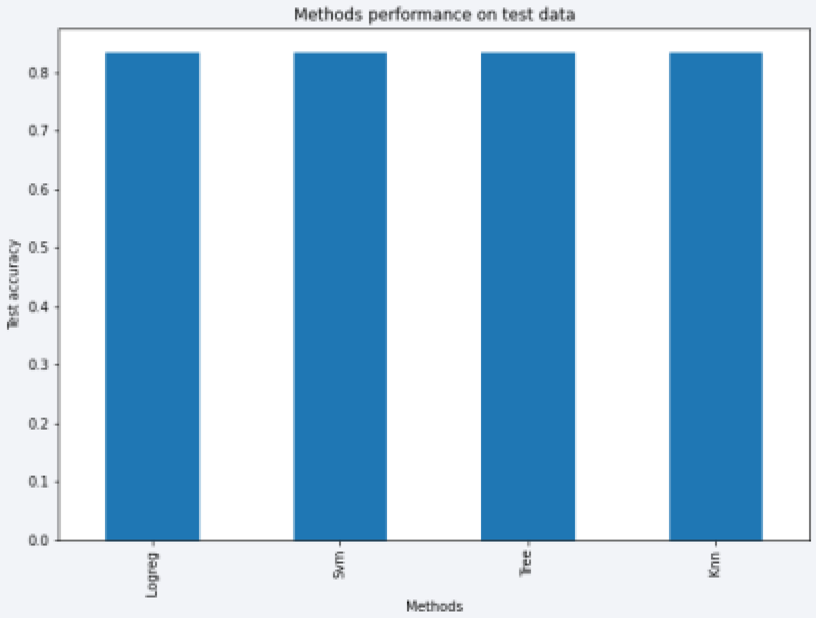
Predictive Analysis (Classification)

Classification Accuracy

```
tuned hyperparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'splitter': 'random'}
```

	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333

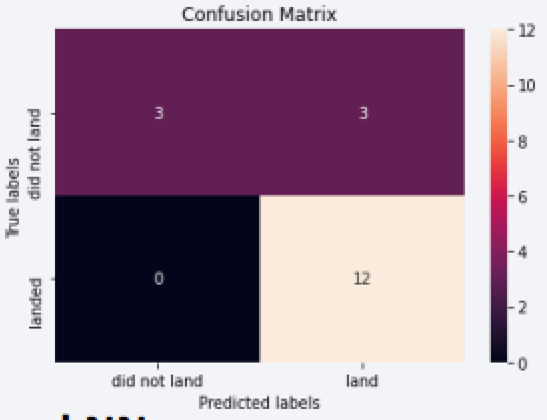




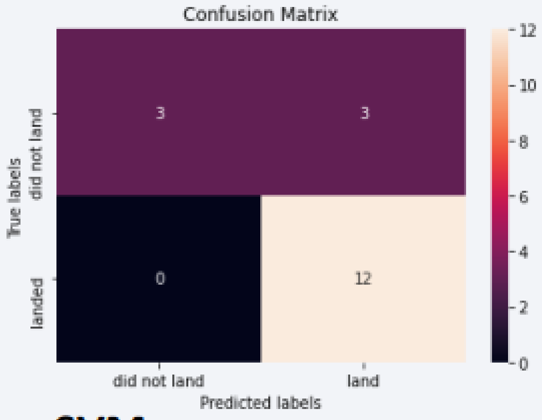
For accuracy test, all methods performed similar.
We could get more test data to decide between them. But if we really need to choose one right now, we would take the decision tree.

Confusion Matrix

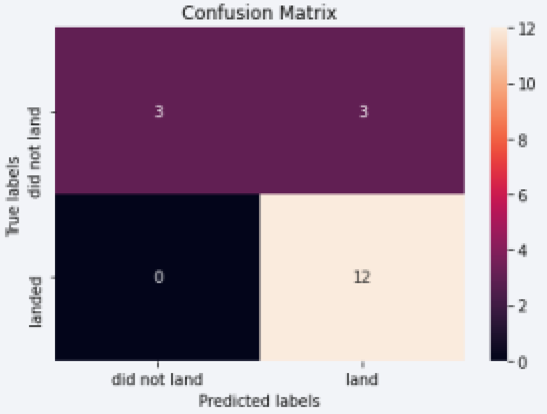
Logistic regression



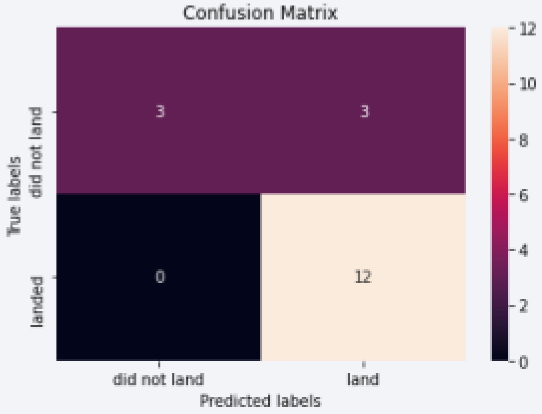
Decision Tree



kNN



SVM



		Actual values	
		1	0
Predicted values	1	TP	FP
	0	FN	TN

As the test accuracy are all equal, the confusion matrices are also identical. The main problem of these models are false positives.

Conclusions

- The success of a mission have relation with the launch site, the orbit and the number of previous launches.
- There has been a gain in knowledge between launches that allowed to go from a launch failure to a success.
- The orbits with the best success rates are GEO, HEO, SSO, ES-L1.
- Depending on the orbits, the payload mass can be a criterion to take into account for the success of a mission.
- Some orbits require a light or heavy payload mass. But generally low weighted payloads perform better than the heavy weighted payloads.
- KSC LC-39A is the best launch site but we cannot explain why. We need more atmospheric data to make conclusions.
- The best model for these data set is the Decision Tree Algorithm because it has a better train accuracy

Appendix

Thank you!